



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**ON-LINE TECHNICKÝ DENÍK**

ON-LINE TECHNICAL DIARY

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN JAROŠ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. VLADIMIR BARTÍK, Ph.D.**

**BRNO 2018**

## **Zadání bakalářské práce**

Řešitel: **Jaroš Martin**

Obor: Informační technologie

Téma: **On-line technický deník**

**On-Line Technical Diary**

Kategorie: Informační systémy

### **Pokyny:**

1. Seznamte se s principy tvorby webových aplikací.
2. Analyzujte požadavky na webovou aplikaci, která bude sloužit jako technický zápisník motorového vozidla. Uživatel bude moci přidat libovolný počet různých typů vozidel, u každého bude moci sledovat vývoj spotřeby vozidla, a dále různé výdaje a opravy, na jejichž nutnost bude uživatel upozorňován např. e-mailem (konec STK, výměna oleje apod.)
3. Dle požadavků navrhnete aplikaci. Pro rychlejší načítání dat navrhnete způsob agregace dat z databáze.
4. Navrženou aplikaci implementujte a ověřte její funkčnost na vhodném vzorku dat.
5. Zhodnoťte dosažené výsledky a další možné pokračování tohoto projektu.

### **Literatura:**

- Welling, L., Thomsonová, L.: PHP a MySQL: rozvoj webových aplikací. Vyd. 1. Praha: SoftPress, 2003, 910 s. ISBN 80-86497-60-7.
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015. ISBN: 978-80-251-4573-9

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## Abstrakt

Práce se zabývá tvorbou webové aplikace, která slouží jako technický deník motorových vozidel. Aplikace umožňuje registraci běžného nebo firemního uživatele, který může přidat libovolný počet vozidel. U firemního účtu má uživatel možnost přiřadit správu vozidla konkrétní osobě. Přidaná vozidla disponují možností sledovat vývoj spotřeby a nákladů na údržbu. Dále aplikace umožňuje správu důležitých upozornění týkajících se vozidla se systémem včasného varování před vypršením expirace.

## Abstract

This thesis is about designing a web application, which works as a technical diary of motor vehicles. The application allows you to register regular or company user who can list any number of vehicles. For company's account its user has the option to assign vehicle's administration to a specific person. Listed vehicles have the ability to monitor its consumption and maintenance costs development. Furthermore, the application allows you to manage important vehicle notifications with its early warning system before expiration.

## Klíčová slova

Informační systém, Technický deník, Databáze, Webová aplikace, PHP, HTML, CSS, Javascript

## Keywords

Information system, Technical Diary, Database, Web application, PHP, HTML, CSS, Javascript

## Citace

JAROŠ, Martin. *On-line technický deník*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Vladimír Bartík, Ph.D.

# On-line technický deník

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vladimíra Bartíka, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Jaroš  
15. května 2018

## Poděkování

Rád bych poděkoval svému vedoucímu Ing. Vladimíru Bartíkovi, Ph.D. za vedení, podporu a rady, které mi poskytoval při tvorbě. Také bych rád poděkoval všem, kteří mi dali cennou zpětnou vazbu a hlavně své přítelkyni, za její věčné názory a konstruktivní kritiku.

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>3</b>  |
| <b>2</b> | <b>Teorie pro vývoj informačního systému a webové aplikace</b> | <b>4</b>  |
| 2.1      | Informační systém . . . . .                                    | 4         |
| 2.2      | Použité technologie . . . . .                                  | 4         |
| 2.2.1    | HTML . . . . .   | 5         |
| 2.2.2    | CSS . . . . .  | 5         |
| 2.2.3    | Javascript a jQuery . . . . .                                  | 6         |
| 2.2.4    | PHP . . . . .  | 6         |
| 2.2.5    | Databáze MySQL . . . . .                                       | 7         |
| <b>3</b> | <b>Analýza požadavků</b>                                       | <b>8</b>  |
| 3.1      | Správa účtů . . . . .  | 8         |
| 3.2      | Správa vozidla . . . . .                                       | 9         |
| 3.3      | Evidence spotřeby . . . . .                                    | 10        |
| 3.4      | Evidence oprav . . . . .                                       | 10        |
| 3.5      | Evidence upozornění . . . . .                                  | 11        |
| 3.6      | Správa nastavení odesílání statistik a upozornění . . . . .    | 11        |
| <b>4</b> | <b>Návrh</b>   | <b>13</b> |
| 4.1      | Architektura systému . . . . .                                 | 13        |
| 4.1.1    | Klient . . . . .   | 13        |
| 4.1.2    | Server . . . . .   | 14        |
| 4.1.3    | Databáze . . . . .   | 14        |
| 4.2      | Návrh databázového schématu . . . . .                          | 14        |
| 4.2.1    | Uživatelé . . . . .  | 14        |
| 4.2.2    | Vozidla . . . . .  | 14        |
| 4.2.3    | Sdílená správa vozidla . . . . .                               | 15        |
| 4.2.4    | Statistiky a upozornění . . . . .                              | 16        |
| 4.2.5    | Agregace dat . . . . .   | 17        |
| 4.3      | Návrh uživatelského rozhraní . . . . .                         | 17        |
| 4.3.1    | Nepřihlášený uživatel . . . . .                                | 17        |
| 4.3.2    | Přihlášený uživatel . . . . .                                  | 18        |
| <b>5</b> | <b>Implementace</b>  | <b>21</b> |
| 5.1      | Použité nástroje a adresářová struktura . . . . .              | 21        |
| 5.1.1    | Nástroje . . . . .   | 21        |
| 5.1.2    | Adresář . . . . .  | 21        |

|          |   |           |
|----------|---|-----------|
| 5.2      | Implementace částí systému . . . . .      | 22        |
| 5.2.1    | Navázání spojení s databází . . . . .     | 22        |
| 5.2.2    | Uživatelský účet . . . . .                | 22        |
| 5.2.3    | Přidání nového vozidla . . . . .          | 24        |
| 5.2.4    | Zobrazení vozidla . . . . .               | 24        |
| 5.2.5    | Přiřazení vozidla . . . . .               | 26        |
| 5.2.6    | Odeslání upozornění a statistik . . . . . | 26        |
| <b>6</b> | <b>Testování</b>                          | <b>28</b> |
| 6.1      | Automatizace . . . . .                    | 28        |
| 6.2      | Uživatelské testování . . . . .           | 29        |
| <b>7</b> | <b>Závěr</b>                              | <b>31</b> |
|          | <b>Literatura</b>                         | <b>32</b> |
| <b>A</b> | <b>Obsah CD</b>                           | <b>34</b> |
| <b>B</b> | <b>Testovací protokol</b>                 | <b>35</b> |
| <b>C</b> | <b>Náhled aplikace</b>                    | <b>36</b> |

# Kapitola 1

## Úvod

Běžní uživatelé motorového vozidla, kteří své auto využívají na každodenní cestu do práce, školy nebo jen na občasné cesty za rodinou či na výlet, by měli mít přehled o vozidle, které používají. Ať už jde o informace ohledně poslední výměny oleje nebo data vypršení platnosti STK. Takové případy se mohou občas značně prodražit, a to buď pokutou za propadlou STK nebo opravou zadrhnutého motoru.

Trh nabízí několik možností řešení tohoto problému. Ať už to je formou fyzických poznámek například v kalendáři, které však mají svá omezení a uložená data se mohou snadno ztratit, nebo virtuálních aplikací pro chytré telefony či webové aplikace. Tato forma by nastíněný problém dokázala téměř vyřešit, avšak v drtivé většině případů se jedná o aplikace zastaralé nebo neudržované. Mnohé jsou v cizím jazyce, neumožňující přidat upozornění nebo jsou příliš složité a nepřehledné. Navíc téměř žádná z nich nedovoluje správu firemních aut.

Cílem bakalářské práce je navrhnout a vytvořit informační systém, který dokáže spojit nejdůležitější aspekty správy aut pro jednotlivce a nebo pro auta firemní. Mezi nejdůležitější možnosti patří: sledování vývoje spotřeby, evidence oprav, přidávání upozornění s určitou dobou expirace a následné přehledné zobrazení statistik. Hlavním rozdílem oproti konkurenci je firemní správa vozidel, která bude umožňovat sdílení vozidla zadaného ve firemním profilu s uživatelem, který ho bude běžně používat.

Tato práce napřed v kapitole 2 seznámí čtenáře s teorií použitých technologií pro vytvoření informačního systému. Další kapitola 3 se bude věnovat analýze požadavků, které budou kladeny na aplikaci. Kapitola 4 bude navazovat na předešlou kapitolu a zaměří se na podrobnější zpracování požadavků, včetně ER diagramů pro zobrazení vztahů. Předposlední kapitola 5 rozebírá podrobněji fungování nejdůležitějších částí práce včetně ukázek uživatelského rozhraní. Poslední kapitola 6 patří testování. Budou zde nastíněné techniky testování aplikace během implementací a manuálního testování.

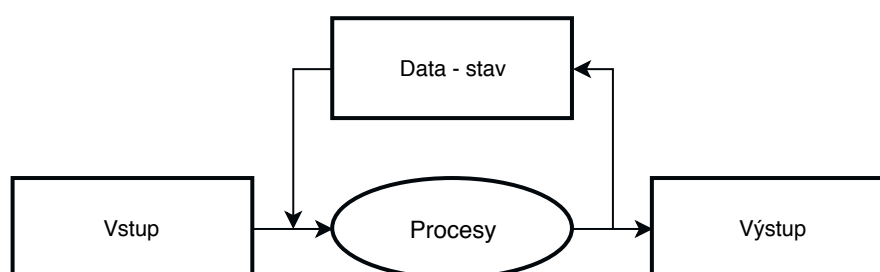
## Kapitola 2

# Teorie pro vývoj informačního systému a webové aplikace

Ještě před analýzou požadavků na aplikaci zde bude popsáno několik základních informací ohledně informačního systému jako celku. Dále bude popsáno jaké technologie jsou aplikovány pro vývoj webové aplikace.

### 2.1 Informační systém

Informační systém, dále jen IS, může být chápán jako systém pracující s informacemi. Jak může být viděno na obrázku 2.1 skládá se ze vstupní části, kde jsou do systému vkládány dané informace. Poté probíhá transformace nad daty, resp. informacemi ze vstupu, pomocí různých algoritmů. Výstup z IS nemusí záviset na jednotlivých vstupech, ale na celkovém stavu IS. [3]



Obrázek 2.1: Schéma informačního systému

### 2.2 Použité technologie

V téhle části budou popsány základní technologie použité pro vytvoření webové aplikace. Hned na začátku bych rád zmínil, že jsem nepoužil žádný z běžných frameworků pro tvorbu, jelikož jsem chtěl k práci přistupovat formou učení. Z toho důvodu jsem se snažil všechny části systému implementovat bez závislosti na jiných částech.



### 2.2.1 HTML

HTML neboli *HyperText Markup Language* je značkovací jazyk, který je určen k vytváření dokumentů, obsahujících hypertextové odkazy<sup>1</sup> a pokročilejší formátování.

#### Historie

Původ HTML sahá až do konce 80.let. Tamní výzkumníci chtěli začít sdílet dokumenty o výzkumech pomocí sítě, aby si tak usnadnili práci. Důležitou roli tady hráli Tim Berners-Lee a Robert Cailliau, kteří pracovali na propojeném informačním systému. Tim Berners-Lee poté v roce 1990 představil jazyk HTML spolu s prvním webovým prohlížečem *WorldWideWeb*. Roku 1992 představili web uživatelům internetu a tím rozpoutali revoluci v jeho vývoji. [6]

#### Použití

Jak již bylo řečeno, HTML je značkovací jazyk. Jako značky používá `< text >`. Textu uvnitř těchto značek se říká instrukce, která říká, jaká *akce* se provede.

Značky, neboli *tagy*, se dělí na párové a nepárové s jejichž pomocí se může upravovat nebo přetvářet obsah webové stránky.

Samotné vytvoření webové stránky spočívá ve vhodném seřazení tagů. Dokument začíná značkou `<html>`, poté následuje definice *hlavičky*, jenž je označena pomocí `<head>` a `</head>`. Hlavičku tvoří metadata, které definují například kódování stránky, titulek nebo i kaskádové styly (o nichž bude řečeno více v další části). Hlavní část, která je ohraničena `<body>` a `</body>`, tvoří samotné tělo stránky, kde se definuje její čtený obsah. Konec dokumentu patří značce `</html>`.

### 2.2.2 CSS

CSS neboli kaskádové styly (z anglického slova *Cascading Style Sheets*)

#### Historie

Před zavedením CSS se používali k formátování stránek pomocí tagů (které byli popsány výše). Například pro písmo kurzívou dvojice značek `<i>` `</i>` nebo centrování textu na střed dokumentu značkami `<center>` `</center>`. Samotná kostra se vytvářela běžně pomocí tabulek.

S postupným zvětšováním dokumentu se obsah stával nepřehledným. Dalšími problémy byly například: špatná kompatibilita mezi prohlížeči nebo pomalé načítání webových stránek kvůli opakujícím se značkám. Tyhle problémy odstraňuje formátování pomocí CSS. Umožňuje vytvořit čistý HTML dokument. [1]

#### Použití

Z popisu výše lze vidět, že CSS je nedílnou součástí HTML stránky. Může být zapsán buďto přímo do značek, kde se přidá atribut `style=""` nebo do hlavičky HTML dokumentu pomocí dvojice značek `<style>` `</style>`.

---

<sup>1</sup>Nelineární text obsahující odkazy

Nejpoužívanější metodou je však zapisování do externího souboru s koncovkou `.css`, na který se v hlavičce stránky odkazuje<sup>2</sup>. Obsah stránky je potom tvořen pomocí minimálně tří pravidel. Prvním je určení *selektoru*, sloužícím pro identifikaci prvků na stránce.<sup>3</sup> Dalším je dvojice název vlastnosti a hodnoty, která určuje jednotlivá pravidla. Pravidel může být několik, a ty jsou jednotlivě odděleny středníkem.

### 2.2.3 Javascript a jQuery

Javascript může být chápán jako jeden ze základních součástí každé webové aplikace. Přidává možnosti interaktivity stránek nebo rovnou tvoří celé aplikace.

Jedná se o objektově orientovaný jazyk se syntaxí podobnou C,C++<sup>4</sup> a Java<sup>5</sup>, který je většinou spouštěn na straně klienta (existují i řešení pro použití na straně serveru<sup>6</sup>).

#### Historie

V hodně zkrácené verzi, se za zrození Javascriptu může vděčit konkurenčnímu boji mezi společnostmi Netscape<sup>7</sup> a prohlížečem Internet Explorer, kdy programátor Brendan Eich dostal za úkol naprogramovat odlehčenou verzi Javy. Stihl to za 10 dnů a tím odstartoval novou éru Javascriptu. [5]

#### Použití

Podobně jako CSS se Javascript může psát přímo do dokumentu pomocí dvojice značek `<script>` `</script>` a nebo zapisovat do externího souboru s koncovkou `.js`, na který se ve stránce odkáže. Ve většině případů jsou definovány Javascriptové funkce, ke kterým je přistupováno pomocí specifické události<sup>8</sup> v HTML tagu, nebo se píše kód přímo do políček atributů(*události*) `<tag událost='kód'>`.

#### jQuery

jQuery je jedna z knihoven Javascriptu. Díky filozofii, která klade důraz na jednoduchost a rychlost se brzy stala nejpoužívanější knihovnou Javascriptu.

Hlavním kamenem úspěchu je způsob a práce s elementy v HTML dokumentu, neboli práce s DOM<sup>9</sup> nad nimiž se budou provádět operace.

### 2.2.4 PHP

PHP (původně *Personal Home Page*) je rekurzivní akronym<sup>10</sup> *PHP: Hypertextový Preprocessor*. Díky svému všeobecnému využití a všestrannosti je velmi rozšířený. [4]

---

<sup>2</sup>Ukázka způsobu spojení CSS a dokumentu [https://www.w3schools.com/html/html\\_css.asp](https://www.w3schools.com/html/html_css.asp)

<sup>3</sup>Seznam dostupných selektorů [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

<sup>4</sup><http://cppreference.com>

<sup>5</sup><http://www.oracle.com/technetwork/java/index.html>

<sup>6</sup>Jedno z nejznámějších řešení je Node.js <https://nodejs.org>

<sup>7</sup>Roku 2003 odkoupena společností AOL <https://aol.com>

<sup>8</sup>Seznam událostí spouštějící Javascript [https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)

<sup>9</sup>Document object model - může být pochopen jako objektově orientovaná prezentace HTML dokumentů <https://www.w3.org/DOM>

<sup>10</sup>Druh zkratky, která je tvořena tak, že ve svém plném znění odkazuje zase na sebe. Tzn. **PHP** -> **Php**: **H**ypertext **P**reprocessor

Svou syntaxí je obdobná jako u jazyků C, Perl a Java. Charakteristický je díky umístění dolaru "\$" před každou proměnou. Podporuje mnoho knihoven, a to zejména pro práci s databází MySQL, což umožňuje tvoření dynamických stránek.

## Historie

Svůj původ nachází v roce 1994, kdy dánský programátor Rasmus Lerdorf vytvořil první část pro osobní domácí stránky<sup>11</sup>. Postupem let docházelo k vylepšování původní verze. V roce 2000 došlo ke kompletnímu přepsání PHP Parseru a dostal označení PHP 4. [4]

## Použití

Jedná se o skriptovací jazyk určený primárně pro programování na webu. Scripty běží na straně serveru a k uživateli se dostane až samotný výsledek. PHP může být vloženo do HTML dokumentu na libovolné místo pomocí značek `<?php ?>`, které budou obsahovat kód.

### 2.2.5 Databáze MySQL

Posledním základním kamenem dynamické webové stránky, který tady bude popsán je databázový systém MySQL. MySQL je systém pro řízení relačních databází umožňující ukládat obrovská množství dat. Díky apelování na rychlost zobrazení dat již od samotného začátku jeho vývoje a snadnému zprovoznění, se systém brzy dostal do velké obliby zejména u programátorů webových stránek.

Stejně jako pro ostatní relační databáze využívá pro komunikaci jazyk SQL<sup>12</sup>. Díky tomu dovoluje vytvářet, přidávat, odebírat či měnit databáze a jejich obsah. [2]

Pro správu databáze může být využit phpMyAdmin<sup>13</sup>, což je PHP aplikace, která je dostupná ve webovém rozhraní a umožňuje spravovat MySQL databáze.

Pracovat s databází se může pomocí jazyka PHP, který byl stručně popsán výše. Přístupeno k ní může být například pomocí rozšíření PDO<sup>14</sup> a nebo MySQLi<sup>15</sup>, která jsou součástí PHP verze 5. V práci bude použit s rozšířením PDO.

---

<sup>11</sup>Proto původní název nalézá v *Personal Home Page*

<sup>12</sup>Structured Query Language - strukturovaný dotazový jazyk

<sup>13</sup><https://www.phpmyadmin.net>

<sup>14</sup>rozšíření PHP pro komunikaci s databázovým serverem <http://php.net/manual/en/book.pdo.php>

<sup>15</sup><http://php.net/manual/en/book.mysqli.php>

## Kapitola 3

# Analýza požadavků

V této kapitole budou vysvětleny požadavky, které jsou kladeny na informační systém pro *Technický deník motorových vozidel*. Taktéž se u jednotlivých požadavků rozeberou důležitá specifika na implementaci. Jednotlivými částmi se uvažuje: správa účtu a vozidla včetně jeho přehledu, evidence spotřeby, oprav, upozornění a spolu s jejich systémem varování a správou statistik.

Cílem aplikace je vytvoření webového rozhraní, které bude usnadňovat jednotlivým uživatelům nebo firmám správu motorových vozidel. Hlavním aspektem je správa samotného vozidla, které je uvedeno v *garáži* spolu s ostatními vozidly daného uživatele. V případě firemního typu účtu<sup>1</sup>, je dovolena správa vozidel více uživatelům. U každého z aut bude umožněno vést evidenci spotřeby, oprav a spravovat upozornění na blížící se důležitou událost (například konec STK). V systému nesmí chybět automatická kontrola upozornění a zaslání statistik na e-mail.

Nyní budou popsány a blíže rozvedeny důležité součásti informačního systému:

- **Správa účtu**
- **Správa vozidla**
- **Evidence spotřeby**
- **Evidence oprav**
- **Evidence upozornění**
- **Správa nastavení odesílání statistik a upozornění**

### 3.1 Správa účtů

Správa účtů je základním prvkem systému. Bez účtu by uživatel nemohl s aplikací jakkoliv manipulovat. Na obrázku 3.1 je znázorněn diagram případu užití účtů, který bude následně popsán.

Uživatel by se do systému měl dostat přihlášením nebo v případě prvního navštívení pomocí registrace a následným přihlášením. Při registraci musí systém vyzvat uživatele na vyplnění několika povinných základních údajů včetně kontaktního e-mailu a zvolení typu účtů. Pokud je zvolen firemní účet bude se muset vyplnit rovněž název firmy.

---

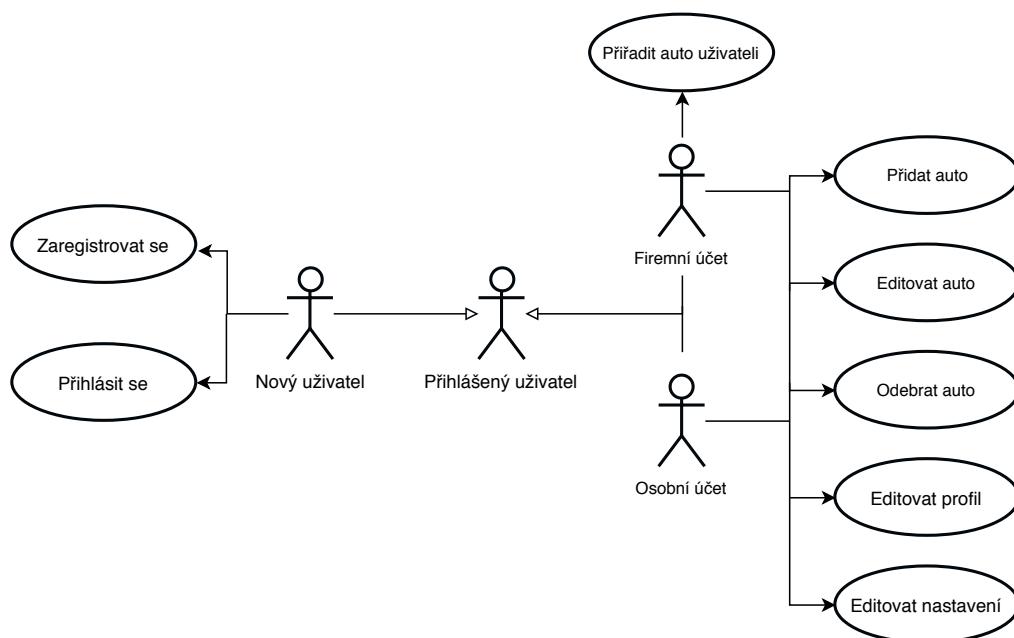
<sup>1</sup>Typ účtů si může uživatel zvolit při registraci

Po přihlášení se očekává stručný přehled aut v hlavním přehledu, neboli *garáži*. Pokud zde nebude přidán žádné vozidlo, nebo naopak se bude požadovat přidání dalšího, aplikace poskytne intuitivní možnost na jeho přidání. U jednotlivě zobrazených aut se budou vyskytovat důležité informace, jako jsou:

- Název auta a značku, aby bylo možné auto, případně více aut jednoduše identifikovat.
- Statistiky spotřeby a oprav za poslední půlrok.
- Upozornění, které se blíží ke konci své doby platnosti.

Samotné informace o účtu budou zobrazeny na profilové stránce, která bude rovněž obsahovat možnosti změny stávajících informací. Dále se zde budou zobrazovat bezpečnostní informace, konkrétněji:

- Vizualizace síly použitého hesla pro přihlášení s možností jeho změny.
- Evidence provedených přihlášení za poslední měsíc.



Obrázek 3.1: Diagram případu užití účtu.

## 3.2 Správa vozidla

Správa vozidla patří mezi nejdůležitější požadavky na systém. Aby bylo možné přidat nové auto musí se napřed provést přihlášení. Tato část již byla popsána v bodě výše. Nyní budou nastíněny specifické požadavky na správu vozidla, které jsou zobrazeny v diagramu 3.1. Hlavními požadavky proto budou: přidání, úprava a odebrání vozidla. Ke každému z požadavků se vztahují důležitá specifika popsána v odstavcích níže.

- **Přidání nového auta:** ve formuláři bude systém požadovat vyplnění několika povinných informací (obdobně jako tomu je při registraci nového uživatele) týkajících se

auta, jako je například název, značka a typ. Také musí umožňovat vyplnění několika nepovinných údajů, ale zato důležitých pro správu. Mezi ně je zařazen obsah motoru, stav tachometru, případně popis a obrázek auta.

- **Editaci auta:** pokud některé z údajů vyplněných při přidání auta nebudou již aktuální nebo budou chybět, bude uživatel požadovat možnost spravovat tyto informace. Aplikace proto bude umožňovat editaci všech dosavadních informací a to včetně přidání, změny nebo odebrání obrázků auta.
- **Odebrání auta:** může se stát, že spravované auto nebude k dispozici nebo se prodá. Z tohoto důvodu bude aplikací poskytnuta volba smazání auta včetně všech jeho údajů. V tomto kroku musí být zajištěna určitá úroveň ochrany s ujištěním uživatele, zda-li má být vozidlo skutečně nenávratně vymazáno ze systému.
- **Přiřazení auta uživateli:** tahle možnost se ukáže pouze pokud v případě firemního účtu. Požadavky na tuhle volbu vyplývají ze situace, kdy je požadována správa firemních aut. Avšak samotné vedení údajů o autě, které uživatel nebude osobně používat je nemyslitelné. Proto aplikace má poskytnout možnost přiřazení daného auta uživateli, který je již zaregistrovaný v systému a po přiřazení auta se mu auto zobrazí v přehledu. Nebo v případě neregistrovaného uživatele, který obdrží pozvánku na e-mail. E-mail bude zároveň sloužit jako jeho jednoznačný identifikátor.

### 3.3 Evidence spotřeby

Každé motorové vozidlo má svou určitou spotřebu paliva, proto je tahle funkcionality důležitou součástí aplikace.

Po aplikaci bude požadováno, aby si dokázala pamatovat data ohledně spotřeby, které se do ní budou vkládat. Nutností je také vhodné zpracování a zobrazení těchto dat do čitelné podoby.

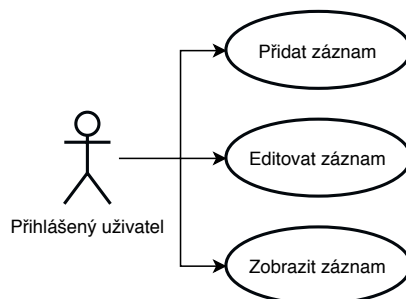
Po každém natankování pohonných hmot do vozidla bude požadováno jejich zaznamenání. Podle obrázku 3.2, který znázorňuje diagram případů užití evidence dat lze vidět, že aplikace bude dávat možnost přidání záznamů o spotřebě, který bude obsahovat:

- Natankované množství paliva.
- Cena za palivo.
- Počet najetých kilometrů nebo aktuální hodnotu na tachometru.

Po zaznamenání bude požadováno evidování přidáných dat o spotřebě. Aplikace by měla poskytovat vhodné zobrazení těchto informací v podobě statistik pro jednotlivé měsíce. Může se stát, že bude zadána špatná hodnota. Proto je nutné, aby se daly záznamy smazat nebo editovat.

### 3.4 Evidence oprav

Je správné mít pod kontrolou opravy a nečekané výdaje na auto. Tímto může být chápáno i vyměnění kola, dolití oleje do motoru nebo v případě nečekaných výdajů, například prasklé sklo či poškrábaný lak.



Obrázek 3.2: Diagram případu evidence dat.

U aplikace je žádané, aby si dokázala pamatovat případné opravy týkající se vozidla. Podobně jak je tomu naznačeno v diagramu 3.2, bude požadováno vkládání hodnot spolu s jejich vhodným zobrazením ve formě statistik. Vkládaná data budou ve formátu:

- Popis opravy.
- Cena za opravu.
- Datum provedení opravy.

U přidanych záznamů je opět důležitá jejich možnost editace, případně smazání (kvůli možné chybě při přidávání nového záznamu)

### 3.5 Evidence upozornění

Poslední důležitou součástí evidování dat je evidence upozornění. Běžného auta v provozu se týkají materiální i nemateriální části, které mají svou dobu spotřeby nebo expiraci. Nalije-li se do auta olej, který při standardním používání auta vydrží 1 rok, nebo nalepí se na sklo dálniční známka, či udělá se STK. Na všechny tyto části je důležité brát ohled a mít je pod kontrolou.

Proto po aplikaci bude požadován systém umožňující přidat záznam s dobou expirace a následně jeho vhodné zobrazení. Při přidání záznamu bude k dispozici několik z nejčastějších typů upozornění jako je: olej, STK, dálniční známka a lékárnička. V případě nestandardního typu záznamu, poskytne aplikace možnost zapsat jeho název. Dále bude následovat přidání doby expirace a případně doplňující popis.

Pro lepší představu pro uživatele o aktuálním stavu upozornění, bude systém také vizuálně zobrazovat dobu expirace a počet dní do konce.

### 3.6 Správa nastavení odesílání statistik a upozornění

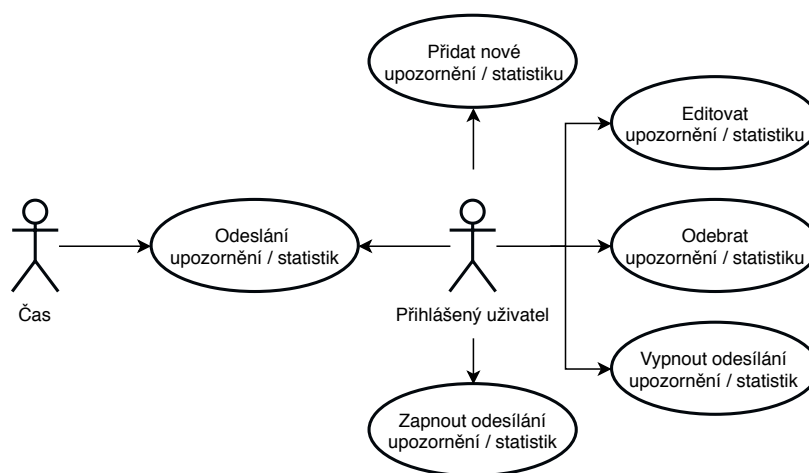
U správného informačního systému, jehož základní funkce bylo popsáno v bodě 2.1, je zpracování vstupních dat. Po aplikaci bude požadováno, aby dokázala zpracovat a vyhodnotit vstupní informace z evidovaných záznamů a na základě jejich hodnot zasílat statistiky a varování na blížící se dobu expirace některého z upozornění.

Na obrázku 3.3, na kterém je popsán diagram případů užití požadované vlastnosti, je možné pozorovat přihlášeného uživatele, kterému se nabízí odeslání upozornění a statistik. Dále je uživateli dostupné přidávat, editovat nebo mazat evidovaná data, která byla popsána

v bodech o evidenci. Důležitou roli bude hrát také čas, podle něhož aplikace vyhodnotí vhodnou dobu na automatické odeslání e-mailu.

K dispozici budou dvě základní položky pro zpracování:

- **Upozornění:** požadovanou vlastností bude zvolení vhodné doby pro odeslání varování na blížící se dobu expirace. Na výběr bude, zda-li má být varování odesláno nebo ne. V prvním případě budou tři možnosti na zvolení frekvence: 3 dny, týden nebo měsíc před jejím vypršení.
- **Statistiky:** podobně jako je tomu u upozornění bude požadováno nastavení doby odeslání statistik. Je předpokládáno, že aplikace nabídne uživateli ze čtyř možností: měsíčně, čtvrtletně, pololetně nebo neodesílat statistiky vůbec.



Obrázek 3.3: Diagram případu užití nastavení odesílání statistik a upozornění.



# Kapitola 4

## Návrh

V následující kapitole bude popsán návrh řešení Technického deníku.

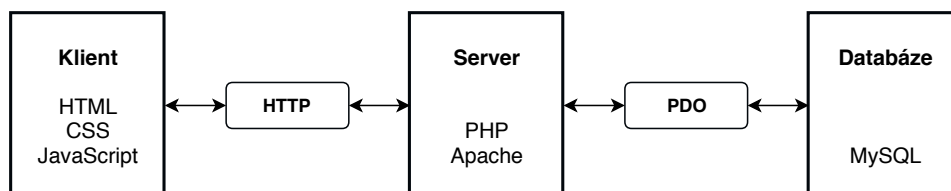
Prvním bodem návrhu bude popis architektury systému, na kterém aplikace poběží. Popis architektury bude rozdělen do tří částí, a sice serverovou, klientskou a databázovou část.

V další části bude blíže vysvětlen návrh databázového schématu spolu s konceptuálními ER diagramy<sup>1</sup>, které přiblíží architekturu použitých tabulek v databázi.

Následující bod bude věnován samotné struktuře systému, která vychází z analýzy požadavků zmíněné v kapitole 3. Jednotlivé části struktury včetně návrhu uživatelského rozhraní.

### 4.1 Architektura systému

Architektura systému je promítnuta na obrázku 4.1, jejíž jednotlivé části budou následně blíže vysvětleny.



Obrázek 4.1: Architektura webové aplikace

#### 4.1.1 Klient

Tahle část komunikuje se serverem, kterému posílá požadavky (uživatel přistoupí na stránku) a zobrazuje odpověď, kterou dostane od serveru. Jako klient je brán webový prohlížeč, díky kterému je umožněno vstoupit na stránku a zobrazí se nám HTML kód spolu s CSS a Javascriptem. Tyhle jednotlivé prvky prohlížeč zkompileje do čitelné podoby webových stránek, tak jak jsou známy. Jedná se tedy o vizuální část aplikace, jejíž jednotlivé body uživatelského rozhraní budou detailně rozvedeny v odstavcích níže.

<sup>1</sup>Metoda pro modelování datových entit a jejich vzájemných vztahů

### 4.1.2 Server

Druhá část jádra aplikace je serverová část. V kapitole o teorii pro vývoj informačního systému 2 byl přiblížen jazyk PHP, který také bude hlavním jádrem dané aplikace. Jak již bylo řečeno, PHP provádí akce na straně serveru. To znamená, že při dotazu z klientské strany se požadavek napřed zpracuje a poté se pošle odpověď.

O veškerou logiku v informačním systému se tedy postará jazyk PHP. Další body budou věnovány samotnému popisu navržené logiky pro jednotlivé části aplikace.

### 4.1.3 Databáze

Poslední částí architektury je databáze, která slouží jako perzistentní<sup>2</sup> úložiště dat. Pro ukládání dat aplikace se použije MySQL server zmíněný v podkapitole 2.2.5 a se kterým se bude komunikovat pomocí PDO. Jednotlivé tabulky databáze budou zmíněny při návrhu uživatelského rozhraní.

## 4.2 Návrh databázového schématu

Správný návrh databázového schématu je velmi důležitou součástí. Analýzou požadavků 3 na aplikaci, byly určeny její základní vztahy. Z těch bude vycházeno při tvorbě ER diagramů. Poslední část bude věnována agregacím dat spolu s jejich principem ukládání a zobrazování.

### 4.2.1 Uživatelé

Jelikož bude umožňovat správu vozidel, která bude určena jednotlivým uživatelům, budou napřed navrhovány tabulky sloužící pro správu účtů. Vychází se z ER diagramu na obrázku 4.2. Hlavní entitou jsou zde považováni uživatelé, u kterých se také bude uchovávat historie přihlášení, aby se v budoucnu mohlo zabránit případnému *brute force* útoku na účet. Uživatel bude mít navíc přehled o přihlášení, které provedl. Kromě informací, které jsou spjaté s uživatelem, jako je jméno, typ účtu a případně název firmy, musí uživatel při registraci vyplnit také e-mail. Na zadaný e-mail následně obdrží ověřovací kód. Kód bude rovněž součástí uživatelské entity. Dalšími důležitými prvky entity, následně zmíněnými, je heslo a síla hesla.

### 4.2.2 Vozidla

Dalším požadavkem je uchovávat informace o vozidlech daného uživatele nebo firmy. Hlavní entitní množina zde bude pojmenována *cars*, která uchová všechny potřebné informace o vozidle. Tvořit jí budou povinné atributy, jako jsou název vozidla, značka a model, které musí uživatel zadat při přidávání vozidla. Ačkoliv se nemusejí zdát jako povinné, budou potřebné pro další entitu s názvem *cars\_aggregated*. Tato entita je sloučením základních informací o vozidle, které se budou zobrazovat v přehledu vozidel. Její hlavní funkcí bude v případě firemního typu účtu, propojování správy jednoho vozidla s více uživateli. Tento návrh bude rozveden v bodě 4.2.3.

U každého vozidla bude umožněno uchovávat také jeho obrázky, které budou obsaženy ve vlastní entitní množině.

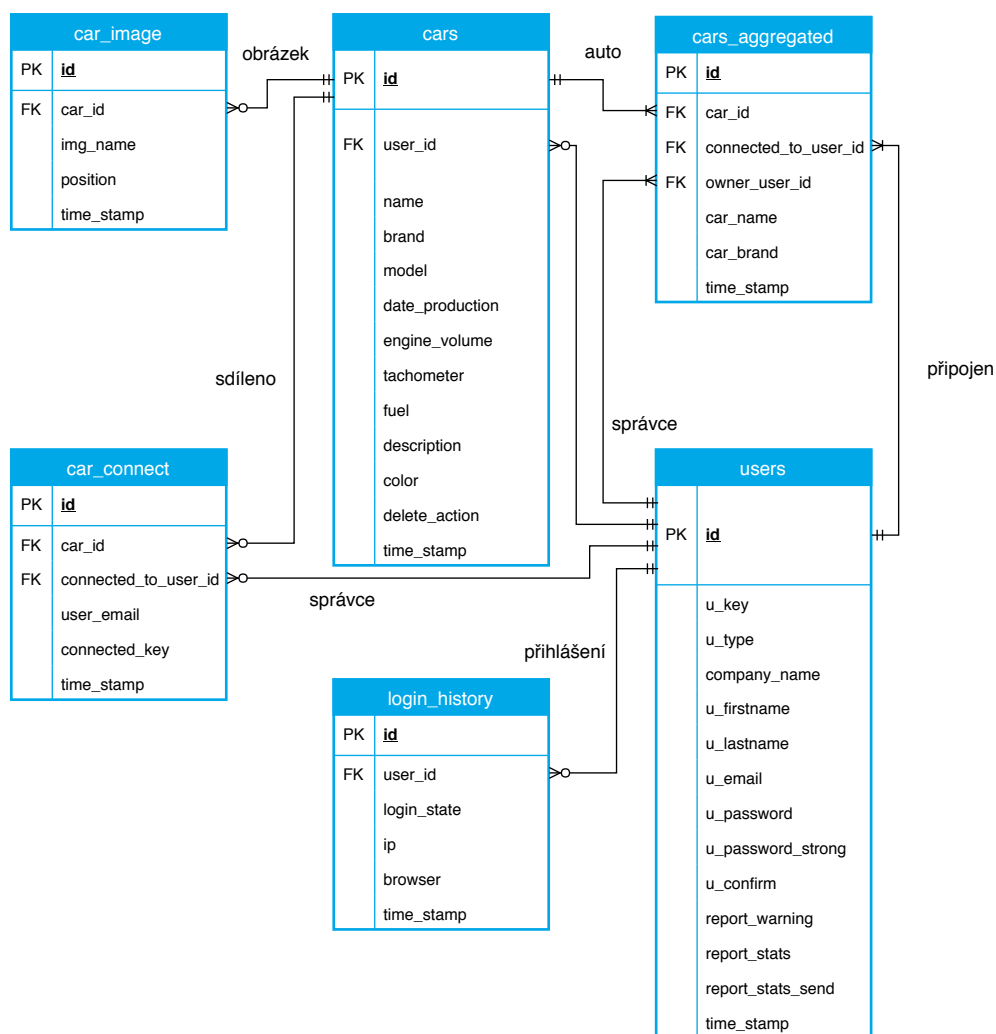
---

<sup>2</sup>Trvalé uložení dat

### 4.2.3 Sdílená správa vozidla

Aplikace bude sloužit také firemní správě vozidel. Je důležité navrhnout vhodný model pro sdílenou správu vozidel. Jako hlavní entitní množinu, zde bude brána *car\_connect*. Při vkládání nového auta bude zvolen e-mail uživatele, na který se mu odešle pozvánka spolu s klíčem pro propojení uživatele s daným autem. Po propojení uživatele a auta, se do entity *car\_connect* přidá identifikační číslo uživatele, tak aby byl přehled o propojených vozidlech s uživateli.

Současně se nově přiřazené auto uloží do tabulky *cars\_aggregated*. Nový záznam bude obsahovat ID vozidla, ID připojeného uživatele, ID vlastníka vozidla a název spolu se značkou. Je důležité, aby tabulka obsahovala i ID vlastníka, protože jen ten bude mít možnost auto smazat ze systému. Připojený uživatel bude mít pouze přístup k autu, které bude mít možnost editovat a vést evidenci, ale smazání vozidla z celého systému mu bude odepřeno. Bude moci pouze odebrat připojené auto ze své garáže.



Obrázek 4.2: ER diagram vztahů správy vozidla, uživatelů a jejich vzájemné propojení

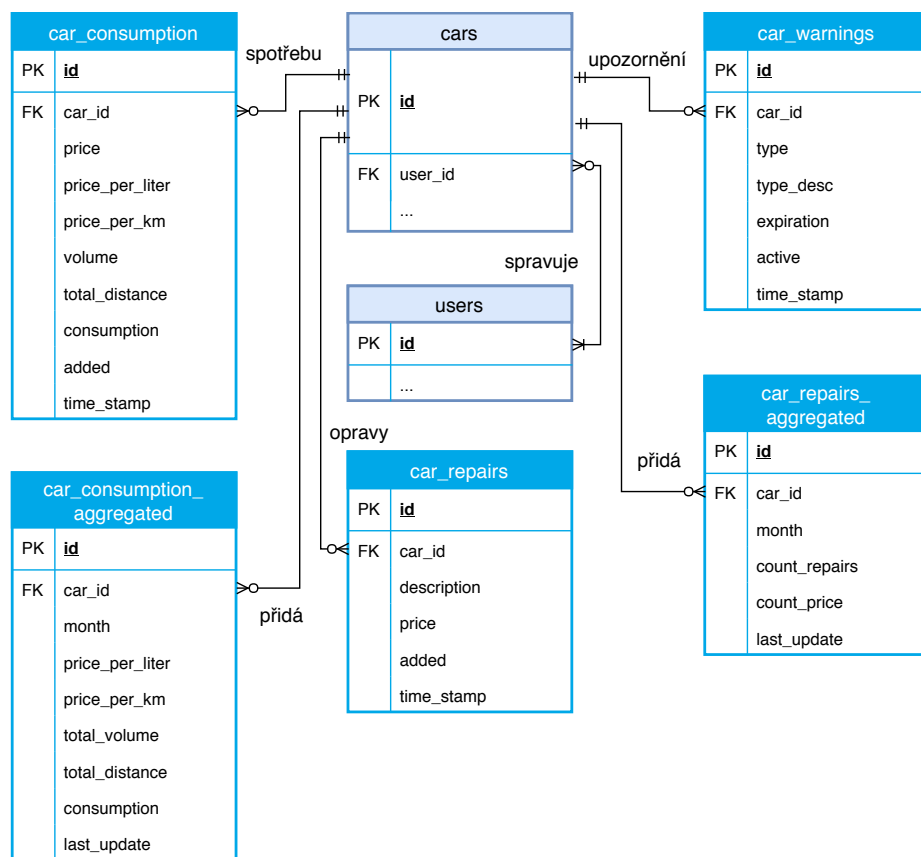
#### 4.2.4 Statistiky a upozornění

Jak již bylo v analýze požadavků řečeno, jádrem aplikace bude evidence statistik, oprav a upozornění. Na obrázku 4.3 je zřetelný diagram vzájemných vztahů v entitách pro evidenci. Jedno vozidlo nemusí mít žádnou zaevidovanou událost a nebo jich může mít několik. Při každém zaevidování nové opravy nebo nových údajů o spotřebě, se v agregačních tabulkách *cars\_consumption\_aggregated* a *cars\_repairs\_aggregated* přepočítají údaje pro každý měsíc zvlášť.

Aplikace eviduje tedy upozornění, u kterých uchovává důležité informace, jako jsou typ upozornění, případně popis. Následně čas expirace a aktivitu. Aktivita značí, zda-li je upozornění aktivní (více informací o aktivitě v kapitole o implementaci).

Dále pokračuje evidence oprav, která ukládá popis opravy, cenu a datum vykonání. Jak již bylo zmíněno, spolu s přidáním opravy se aktualizuje záznam v agregační tabulce. V aktualizovaném záznamu se zvýší číslo opravy spolu se sečtením ceny a uloží se pro daný měsíc.

Posledním požadavkem na evidování je spotřeba. Podobně jako je tomu u evidence oprav, se pro vozidlo ukládá záznam o spotřebě, který obsahuje cenu natankování, cenu za litr, cenu za kilometr, natankované množství, ujetá vzdálenost, spotřeba a datum vložení údajů. Stejným způsobem se spolu s novým záznamem aktualizuje záznam v agregační tabulce. Aktualizace se provede tak, že se například u spotřeby nebo ceny za litr sečtou příslušné hodnoty a vypočte se jejich aritmetický průměr. U dalších se hodnoty pouze sečtou. Následně se uloží k danému měsíci, ke kterému přísluší.



Obrázek 4.3: ER diagram vztahů evidence spotřeby, statistik a upozornění

### 4.2.5 Agregace dat

V podsekcích výše, byly zmíněny agregační tabulky. Jejich hlavní úkol je v uchování zpracovaných dat, aby se nemusely znovu zpracovávat nebo načítat z více tabulek najednou.

První a nejdůležitější tabulkou bude *cars\_aggregated*, která uchovává informace ohledně přiřazených aut k uživateli. Tento krok je důležitý zejména proto, že hlavní přehled aut bude muset zobrazovat také přiřazené firemní auta k uživateli. Aby se při každém načítání přehledu vozidel (garáží) nemusely vybírat data z více tabulek, nahradí tuto operaci načítání dat pouze z jedné tabulky, která si uchovává základní informace o auta, identifikační číslo vlastníka a připojeného uživatele.

Další dvě tabulky *car\_consumption\_aggregated* a *car\_repairs\_aggregated* budou plnit podobnou funkcionalitu. Budou zejména sloužit pro uložení předzpracovaných dat pro každý kalendářní měsíc. Jejich obsah bude sloužit k zobrazení statistik na stránce a statistik odeslaných e-mailem.

## 4.3 Návrh uživatelského rozhraní

Do návrhu aplikace se zahrne také grafická podoba webové aplikace. Budou zde popsány vlastnosti a umístění jednotlivých částí uživatelského rozhraní. Při tvoření jednotlivých částí bude dbáno na zachování intuitivnosti a jednoduchosti celého systému ovládání.

Celý koncept se rozloží pro přihlášeného a nepřihlášeného uživatele. Nepřihlášený uživatel bude mít možnost navštívit pouze obrazovku pro přihlášení, registraci a případně pro kontaktování administrátora, kdyby nastal problém s přihlášením nebo registrací.

Další části obrazovky budou sloužit již pro přihlášené uživatele a zahrnují celou správu.

### 4.3.1 Nepřihlášený uživatel

Obrazovka bude rozčleněna na hlavičku a tělo. Hlavička bude obsahovat v levé části logo, které bude sloužit současně jako odkaz na domovskou stránku. Pravá část bude tvořena dvěma odkazy. Jeden pro návrat na domovskou stránku, druhý pro kontaktní stránku.

Tělo stránky bude tvořeno příslušným obsahem podle situace, kde se právě bude uživatel nacházet. Rozdělení podle situací bude vypadat následovně:

#### Úvodní obrazovka

Tato obrazovka bude obsahovat úvodní titulek, který slouží pro představení stránky novému návštěvníkovi. Pod ním následují odkazy pro registraci a přihlášení.

#### Registrace

Stránka bude tvořena formulářem pro vyplnění několika základních údajů včetně e-mailu a hesla. Bude se zde zobrazovat také výzva pro opakované zadání a síla hesla.

#### Přihlášení

Podobně jako je tomu u registrace, stránka bude obsahovat formulář pro přihlášení. Bude se zde zadávat uživatelský e-mail a heslo.

### 4.3.2 Přihlášený uživatel

Struktura obrazovky po přihlášení se bude lišit od obrazovky před přihlášením a to z důvodů, aby uživatel na první pohled poznal, kde se nachází. Po úspěšném přihlášení se ocitne na přehledu, kde budou k vidění všechna auta v *garáži*.

Obrazovku bude tvořit navigační menu, které bude fixováno na levé straně a obsah stránky. Jednotlivé části a elementy stránky budou rozepsány v následujících bodech.

#### Menu

Bude navigačním bodem umožňujícím se pohybovat po aplikaci. Je rozložené po celé výšce stránky a rozdělené na pět částí.

- **Logo:** první část bude tvořena logem, které současně odkazuje na přehled.
- **Oslovení:** další část bude plnit úkol oslovení přihlášeného uživatele. Uživatel uvidí své jméno a ikonku uživatele. Pokud se bude jednat o firemní účet uvidí ikonku budovy, která má imitovat firmu.
- **Garáž:** třetí část obsahuje tu nejvýznamnější část a sice přehled a odkaz na stránku pro přidání nového vozidla.
- **Správa:** čtvrtou částí bude správa, která je spojená s uživatelem. Bude obsahovat odkazy na profil, nastavení účtu a odhlášení.
- **Ostatní:** poslední část se bude věnovat odkazům na nápovědu a v případě dotazů na kontaktní formulář.

#### Přehled

Obrazovka přehledu bude sloužit pro zobrazení všech aut v garáži. Auta budou situovaná v boxech, které budou sériově řazeny za sebou. V levé části u každého ze zobrazených aut bude viděn jeho název, případně značka a obrázek, tak aby bylo na první pohled zřejmé, o jaké auto se jedná.

Dále podle analýzy požadavků bude umožněno pozorovat statistiky, které k danému vozidlu patří. Pod nadpisem se zobrazí aktuální počet běžících upozornění a datum nejbližší expirace. Napravo od upozornění budou umístěné vedle sebe dva grafy. První bude ukazovat historii oprav a druhý vývoj spotřeby za posledního půl roku.

#### Zobrazení vozidla

Stránka pro zobrazení vozidla se bude skládat z několika sektorů.

- **Statistiky:** do prvního sektoru, který bude sloužit pro informativní účely, jsou zařazeny tři části. První část bude obsahovat dvě blížící se upozornění, kterými končí jejich doba expirace. Další dva budou tvořeny grafy. Ty fungují stejně jako v přehledu vozidel, a sice zobrazují statistiky oprav a vývoje spotřeby za posledního půl roku. Všechny části budou obsahovat v pravém horním rohu tlačítko pro přidání nového záznamu.
- **Profil auta:** po statistikách následuje sektor, který bude obsahovat všechny informace, které byly zadány při vložení auta. Levá část tedy bude tvořena obrázkem auta.

Po pravé straně obrázku budou následovat informace o vozidle a v pravém horním rohu tlačítko pro editaci. Po stisknutí tlačítka se zobrazí okno pro změnu údajů.

- **Galerie:** se bude nacházet po pravé straně od profilu auta. Bude sloužit jako galerie pro dané vozidlo. V pravé části tohoto sektoru bude umístěno tlačítko pro přidání nového obrázku. U každého obrázku se bude zobrazovat možnost smazat nebo učinit ho hlavním obrázkem pro auto.
- **Upozornění:** v dalším sektoru bude umístěna evidence upozornění. Sektor bude rozdělen do dvou částí. Levá část bude obsahovat aktivní upozornění, spolu s jejich typem a dnem expirace. Druhá část bude umístěna napravo, rovněž bude mít tabulku se seznamem všech upozornění. Položky v tabulce budou barevně rozlišeny na aktivní a neaktivní. Uživatel bude mít také možnost jejich stav z neaktivní změnit na aktivní a naopak. Navíc každou položku může odebrat.

V pravém horním rohu bude umístěno tlačítko pro přidání nového upozornění, po jehož stisknutí se zobrazí okno pro přidání nového upozornění.

- **Spotřeba:** následující sektor bude obsahovat evidenci spotřeby. Stejně jak je tomu v sektoru upozornění, bude rozdělen do dvou částí. Levá část bude obsahovat informace o průměrné spotřebě vycházející ze všech zadaných dat. Pravá část bude tvořena tabulkou s evidencí všech zaznamenaných údajů o spotřebě. Každý řádek tabulky se může odebrat.

Nesmí chybět také tlačítko pro přidání nových údajů o spotřebě, které bude situováno v pravém horním rohu.

- **Opravy:** poslední sektor stránky slouží pro evidenci oprav. Opět bude rozdělen na dvě části. Levá bude obsahovat informace o počtu provedených oprav a jejich celkové ceny. Pravou část bude tvořit tabulka s přehledem přidanych oprav.

Přidání nové opravy se provede opět kliknutím do pravého horního rohu na tlačítko, které vyvolá okno, kde se zadá několik údajů o opravě včetně ceny.

### Přidání nového vozidla

Podle analýzy požadavků stránku pro přidání nového vozidla bude tvořit jeden formulář, do kterého se budou vyplňovat údaje o autě. Některé údaje budou povinné, a to název, značka a model. Ostatní dobrovolné, jako je rok výroby auta, obsah motoru, stav tachometru, druh paliva či barva, obrázek nebo popis. U firemního typu účtů je zde přidána možnost přiřadit auto uživateli skrze jeho e-mail.

### Profil

Stránka se správou účtu byla prvním požadavkem na aplikaci. Bude rozdělena do dvou částí, kde první část bude obsahovat informace o účtu, a to: jméno(název firmy), e-mail, datum zaregistrování a ukazatel ověřeného účtu přes e-mail. V případě, že účet nebude ověřen, zobrazí se tlačítko pro odkaz na opětovné zaslání ověřovacího e-mailu. V pravém horním rohu se bude zobrazovat tlačítko pro editaci profilu. Po jeho rozkliknutí se zobrazí okno, které umožní změnit jméno, příjmení (název firmy) nebo e-mail.

Druhou část bude tvořit bezpečnost a přihlášení. Zobrazí se zde ukazatel síly hesla a tabulka s historií přihlášení za jednotlivý měsíc. Tabulka bude obsahovat datum provedeného přihlášení, IP adresu, prohlížeč a ukazatel úspěšného nebo neúspěšného přihlášení.

## Nastavení

Hlavním požadavkem na tuhle stránku bude editace frekvence odesílání upozornění a statistik na e-mail. Bude rozdělena do těchto tří sektorů řazených pod sebou:

- **Odesílání upozornění:** bude obsahovat informaci o aktuálně nastavené frekvenci odeslání upozornění. V pravém horním rohu se budou nacházet dvě tlačítka. První z nich bude umožňovat odeslání aktuálních upozornění, která odpovídají dané frekvenci. Tlačítko bude viditelné pokud bude frekvence nastavená, v opačném případě se tlačítko nebude zobrazovat. Dalším tlačítkem bude editace, která zobrazí okno s možnostmi změny frekvence.
- **Odesílání statistik:** druhým sektorem bude informování o frekvenci odesílání statistik. Stejně jak je tomu v prvním bodě, bude obsahovat dvě tlačítka, která mají taktéž podobnou funkčnost. Po stisknutí prvního z tlačítek se odešlou statistiky ze zadaného období. Druhým tlačítkem se provede editace frekvence odesílání statistik.
- **Správa připojených účtů:** poslední sektor je věnován správě připojených uživatelů. Zobrazí se pouze pokud bude mít uživatel firemní účet.

Bude tvořena dvěma částmi. V první části bude informace o aktuálním počtu propojených uživatelů s firemními auty. Ve druhé části bude umístěna tabulka se seznamem uživatelů, ke kterým je přiřazeno dané auto, stav přiřazení a datum odeslání pozvánky. Dále bude možné přiřazené uživatele smazat.

V pravém horním rohu se bude nacházet tlačítko pro přidání nového propojení uživatele s autem.

## Nápověda a kontakt

Jedna z mála věcí, které nebyly určeny v analýze požadavků je nápověda a kontakt. Přesto jim bude věnován prostor, protože vhodná nápověda a kontaktní formulář by neměl chybět v žádném informačním systému.

- **Nápověda:** stránka bude obsahovat několik návodů spojených s běžným používáním aplikace, které by mohli uživateli dělat potíže. Bude se jednat o statickou stránku jejíž obsah nebude možné editovat uživatelem.
- **Kontakt:** kontaktní formulář bude sloužit jako kontaktní prostředek se správcem systému v případě dotazů. Bude rozdělen na dvě části, kde první část bude obsahovat základní informace o správci spolu s jeho kontaktem.  
Druhá část bude obsahovat kontaktní formulář, kde se bude muset vyplnit jméno, e-mail a případně dotaz, který bude potřeba odeslat.



## Kapitola 5

# Implementace

Hlavní náplní této kapitoly bude popsání implementačních detailů informačního systému podle požadavků z předchozího návrhu. Bude rozdělena do dvou částí. První část bude obsahovat popis použitých nástrojů a adresářové struktury na serverové části systému. Druhá část se bude věnovat podrobnému popisu některých částí systému. Výsledná aplikace bude dostupná z <http://kined.eu>

### 5.1 Použité nástroje a adresářová struktura

Z analýzy požadavků a návrhu byla nadefinována struktura aplikace. Avšak pro její správných chod je potřeba několika nástrojů, které budou popsány níže. Dále bude nastíněna adresářová struktura včetně významů jejích jednotlivých částí.

#### 5.1.1 Nástroje

V kapitole o teorii [2](#) byl určen jako implementační jazyk PHP a databázový systém MySQL. Na téhle dvojici bude postavená celá aplikace. Pro fungování je potřeba nainstalovat na stranu serveru nástroje s podporou těchto technologií. Jedno z řešení, které se nabízí je WampServer <sup>1</sup>, který zajistí nainstalování všech potřebných nástrojů.

#### 5.1.2 Adresář

Adresářovou strukturu tvoří několik částí. Dvě z nich přímo nesouvisí s aplikací, ale jsou její důležitou součástí, a to: adresář obsahující dokumentaci a testy. Další části spolu s jejich stručným popisem jsou zobrazeny níže:

- `\cars` - Adresář obsahující obrázky aut
- `\content` - Adresář s jednotlivými stránkami aplikace
- `\core` - Adresář obsahující důležité inicializační funkce a spouštěcí procedury pro odesílání e-mailů
- `\doc` - Adresář pro dokumentaci
- `\qa` - Adresář s testy

---

<sup>1</sup>Softwarový balíček obsahující Apache, PHP a MySQL <http://www.wampserver.com>

- **\template** - Adresář obsahuje hlavní šablonu stránky včetně obrázků, jQuery a CSS souborů.
- **index.php** - Hlavní soubor, který se spouští při návštěvě stránky.

## 5.2 Implementace částí systému

Tahle část se bude zabývat popsáním implementačních údajů. Ty důležitější části budou rozebrány více technicky. Prvním bodem bude část zabývající se připojením a komunikací s databází, poté práce s uživatelským účtem a následně několik specifických částí.

### 5.2.1 Navázání spojení s databází

Pro práci s databází se využívá PDO, které bylo zmíněné v podkapitole [2.2.5](#). Spojení s databází se musí uskutečnit před každým dotazem, proto je jeho ustanovení jako jeden z prvních inicializačních procesů po vstupu na stránku. Inicializační funkce se nacházejí v adresáři `\core\init.php`. Zde se vytvoří nové instance třídy PDO, která se následně bude při každém dotazu používat.

### 5.2.2 Uživatelský účet

Tato podkapitola je věnována podrobným popisem uživatelských účtů, a sice: registraci, potvrzení účtu a přihlášení.

# Vítejte!

Zaregistrujte si u nás nový účet zdarma.

Typ účtu

Osobní

▼

Jméno

Martin

Příjmení

Jaros

E-mail

martin.jaros@email.cz

Nové heslo

\*\*\*\*\*

Opakovat heslo

\*\*\*\*\*

Síla hesla

Registrovat

Obrázek 5.1: Registrační formulář

## Vytvoření nového účtu

V registračním formuláři se musí jako první zadat typ účtu a v případě firemního typu se zobrazí nové pole *input*, které bude požadovat název firmy. Pole se ukáže díky Javascriptové funkci *show\_company\_name\_check*, která detekuje zvolený typ účtu. Při vyplňování hesla se aktivuje Javascriptová funkce *check\_strong\_password*. Ta přepočítá aktuálně zadanou hodnotu a přiřadí ji body podle jeho obsahu (zda-li obsahuje číslo nebo velké či malá písmena). Po potvrzení se zkontroluje zadaný e-mailem, jestli není zaregistrovaný pod jiným uživatelem. V případě, že v systému neexistuje, přidá nového uživatele do systému. Pokud se uživatel registroval skrze pozvánku, která mu přišla na e-mail, tak se mu rovnou přiřadí auto. Spolu s registrací se uživateli odešle e-mail, který bude obsahovat jednoznačný ověřovací klíč, který se vytvoří pomocí spojení několika funkcí: *md5*<sup>2</sup>, *uniqid*<sup>3</sup> a *rand*<sup>4</sup>, aby se zaručila skutečná jednoznačnost. Spojením těchto tří funkcí je nutné z důvodu *uniqid*, které nemusí vždy generovat náhodné ID (závisí na systémovém čase). Na obrázku 5.1 je vidět jak formulář vypadá.

## Přihlášení

Po zadání přihlašovacích údajů do formuláře, aplikace ověří existenci zadaného e-mailu. V případě že neexistuje, objeví se chybová hláška, v opačném případě se ověří heslo pro uživatele s daným e-mailem. Heslo se ověří díky funkci *password\_verify*, která porovná zadané heslo spolu s hashem u uloženého hesla v databázi. Po ověření se spustí funkce *login\_history\_trigger*, které přidá k danému uživateli záznam o přihlášení spolu s úspěšným nebo neúspěšným pokusem. Aby aplikace rozpoznala, že je uživatel přihlášen musí být v inicializačním souboru, který byl popsán v podkapitole 5.2.1, přidaná funkce *session\_start()*, která inicializuje relaci<sup>5</sup>. Do ní se uloží proměnná, podobně jako do globální proměnné typu pole, a sice `$_SESSION['user_session'] = ID`. Tahle proměnná je potom viditelná všude v aplikaci, kde bude mít inicializační soubor přístup.

- **Brute force:** případný útočník, který se bude snažit o prolomení hesla formou brute force útoku<sup>6</sup> (dále jen BF) bude detekován pomocí funkce *simple\_brute\_force()*. Tahle funkce má za úkol pamatovat si neplatný pokus o přihlášení a v případě zadání tří neplatných pokusů zablokuje přihlášení po dobu 3 minut, tak aby případný útok BF byl znesnadněn. Funguje tak, že si ukládá čas posledního přihlášení z prohlížeče do pole `$_SESSION['BF']`. Pokud pole bude větší jak 3, načte si poslední čas pokusu a přičte k času 3 minuty. Při dalším pokusu se ověří, jestli již zadaný čas uběhl. Po jeho uplynutí, se pole `$_SESSION['BF']` vynuluje a je připraven pro uložení dalších pokusů.

Spolu se zablokováním přihlášení, se uloží do tabulky *login\_history* informace o tom, že z daného prohlížeče s danou IP adresou byly provedeny tři neplatné pokusy o přihlášení k existujícímu uživatelskému e-mailu.

<sup>2</sup>PHP funkce, která vypočítá md5 hash ze zadané hodnoty

<sup>3</sup>PHP funkce pro generování jednoznačného identifikátoru, založená na čase.

<sup>4</sup>PHP funkce, která vygeneruje náhodné číslo

<sup>5</sup>Navázané spojení mezi klientským prohlížečem a serverem, který si uchovává hodnoty spojení daného klienta v dočasném souboru. Soubor se smaže po zrušení prohlížeče, nebo úmyslném zrušení [4]

<sup>6</sup>Forma dešifrování hesla, kde se útočník snaží zadat všechny možné kombinace znaků

## Aktivace účtu

Každý uživatel musí mít aktivovaný účet, jinak nemůže přidat vozidlo. Tahle funkcionality pomůže omezit případné návštěvníky s falešným e-mailem. Neaktivovaný účet upozorní přihlášeného uživatele na jeho stav pomocí červené tečky u profilu v menu. Samotné aktivování účtu probíhá následovně.

Uživatel po registraci obdrží již zmíněný e-mail s klíčem. Po kliknutí na klíč se přesměruje na stránku kam se odešle daný klíč s identifikačním číslem uživatele, dále jen ID. Aplikace pozná díky obsahu adresy, která bude obsahovat *key* a *id*, že se jedná o aktivaci. Poté dotaz přesměruje na stránku *confirm\_user.php*, která podle zadaného ID porovná klíč v adrese s klíčem z databáze. V případě že jsou klíče shodné aktualizuje klíč uživatele v databázi na stav *confirmed*.

### 5.2.3 Přidání nového vozidla

Přidání vozidla funguje obdobně jak je tomu u registrace uživatele. Uživatel vyplní formulář, který se následně odešle na server, který obsah validuje. Oproti registraci zde přibývá možnost propojení uživatele, která bude popsána v další podkapitole 5.2.5 a přidání obrázku vozidla, jehož nahrání bude následně detailněji popsáno.

Možnost přidání obrázku je doplněna o Javascriptovou funkci *show\_image*. Ta zajišťuje validaci vloženého souboru na straně klienta. Kontroluje povolené formáty podle přípony a maximální velikost. Zároveň vybraný obrázek zobrazí vedle formuláře pomocí konstruktoru *FileReader*. Po přidání nového vozidla s obrázkem se ještě jednou provede na straně serveru validace obrázku, kdyby náhodou kontrola u klienta selhala. Následně se vygeneruje pomocí funkce *rand* číselný identifikátor, který bude prezentovat nové jméno obrázku. Výsledný obrázek se nahraje do serverového adresáře podle daných ID:

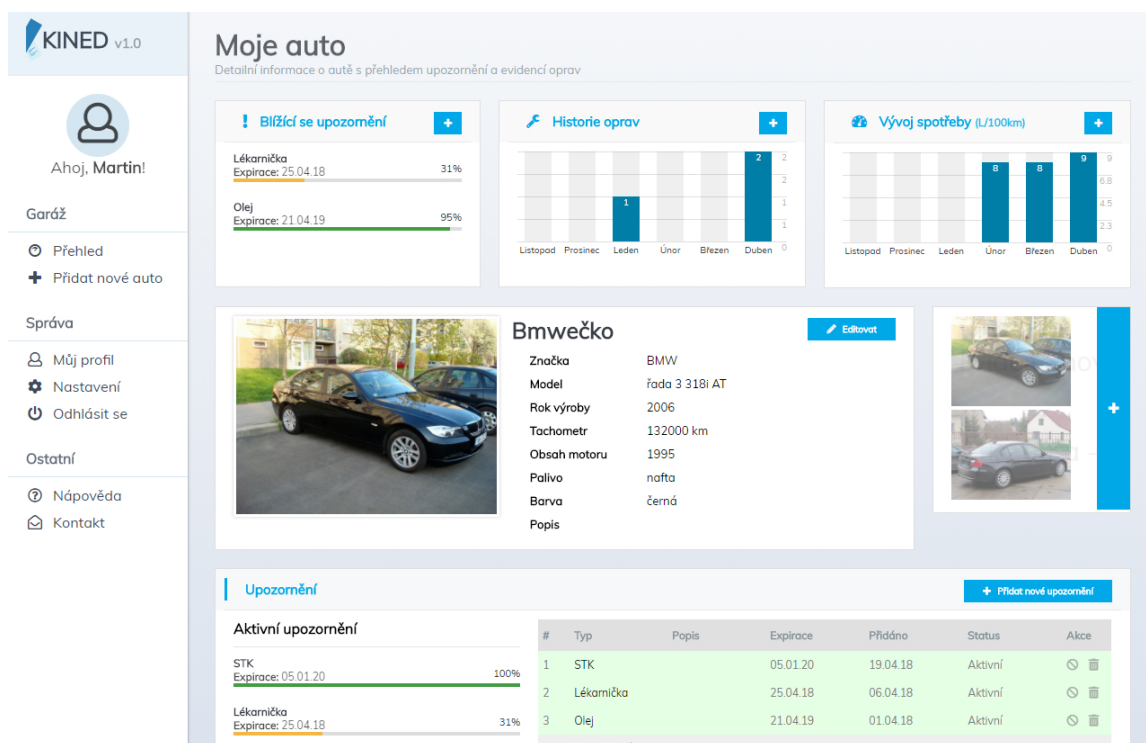
```
\cars\id_uzivatele\id_noveho_auta\id_obrazku.koncovka
```

### 5.2.4 Zobrazení vozidla

Jedná se o jeden z nejkompikovanějších pohledů aplikace. Proto bude rozdělen do tří základních částí, které budou následně podrobněji popsány.

## Grafy statistik

Zobrazování grafů probíhá tak, že data vybírá rovnou z příslušných agregačních tabulek. Napřed se provede určení časového rozsahu, v tomhle případě jsou požadovány statistiky za posledního půl roku. Časový interval se získá spojením funkcí *strtotime* a *date*. *Date* vypíše aktuální den, který se vloží jako parametr do *strtotime*. Do parametru se musí k hodnotě připojit příkaz *first day of -5 month*, společně bude funkce vypadat následovně `strtotime(date("Y-m-d").'first day of -5 month')`; tímto se získá první den, 5 měsíců zpátky. Podobně se určí také první den aktuálního měsíce. Po určení intervalu se vyberou data z tabulek pomocí SQL dotazů, které se odešlou do MySQL databáze prostřednictvím PDO. Aby bylo možné zobrazit i hodnoty v měsících, kde nejsou žádné naměřené hodnoty, musí být vytvořeno pole, které se naplní čísly měsíců z určeného intervalu. Pole měsíců se poté naplní daty z tabulky a zapamatuje si nejvyšší hodnotu (v případě oprav jejich počet, u spotřeby maximální průměrnou hodnotu), aby bylo možné v grafu znázornit vrchol. Graf bude rozložen do 5 částí. Spodní část bude tvořit nula a na vrcholu nejvyšší



Obrázek 5.2: Pohled auta

zaznamenaná hodnota ve všech měsících. Získané data za dané měsíce se zobrazí pomocí percentilu vůči maximální hodnotě.

### Ukazatel průběhu varování

Podobně jako grafy bude detailněji popsáno fungování ukazatele průběhu varování. Cílem této funkčnosti je, aby ukazovala pomocí procent čas do konce expirační doby.

Aplikace si napřed vyžádá data záznamů, kterým neskončila doba platnosti a jsou stále aktivní. Ze získaných dat zjistí dobu přidání a dobu expirace. Pomocí funkce `temp_info_percentil()` vypočítá procentuální hodnotu mezi aktuálním dnem, dnem přidání a koncem platnosti. Zjištěné procento zobrazí funkce `show_bar` jako ukazatel průběhu, jehož barva bude závislá na hodnotě. Nejnovější bude mít barvu zelenou, pak následuje oranžová a červená.

### Přidání nového záznamu

Tahle funkcionalita se zde promítá ve více situacích a liší se jen pár detaily, proto bude popsán jenom společný základ přidání nebo editace záznamu.

Většina dat se přidává pomocí formuláře ve vyskakovacím okně, tzv. *popup*. Popup se zavolá stisknutím příslušného tlačítka pro editaci nebo přidání nového záznamu. Princip ukázky volání popupu je ukázán ve výpisu níže 5.1. Po stisknutí se zavolá Javascriptová funkce `show_edit_car_edit_popup`, která bude obsahovat jako argument ID okna, které se po zavolání otevře. Funkce se postará o identifikaci okna prostřednictvím ID, které bylo doposud skryté a zobrazí ho. Zobrazení se provede pomocí změny CSS stylu z `display: none` na `display: block`. Současně zjistí polohu nového okna vůči aktuálnímu zobrazení

a v případě, že se zde nebude vyskytovat, posune aktuální pohled na zobrazený popup. Součástí otevření okna ztmavne pozadí, tak aby bylo vidět pouze požadované okno.

Popup bude obsahovat formulář, který bude sloužit pro přidání nebo změny údajů. Po potvrzení se data odešlou na server a provede se jejich zpracování. V případě chyby se zobrazí hláška. V opačném případě aplikace vypíše úspěch, okno se uzavře a pohled se aktualizuje.

```
<div id="dialog_1"> Obsah popup okna </div>

<button onclick="show_car_edit_popup('dialog_1')">
    Otevrit
</button>
```

Výpis 5.1: Ukázka výpisu fungování popupu

### 5.2.5 Přiřazení vozidla

Systém poskytuje možnost přiřadit vozidlo libovolnému množství uživatelů, které ho budou spravovat. Přiřazení se může provést již při přidávání nového auta, nebo v sekci nastavení. Ve formuláři se musí zadat e-mail od uživatele a přiřadit příslušné vozidlo. Po potvrzení aplikace zkontroluje jestli již spojení daného e-mailu s uživatelem neexistuje a odešle pozvánku. Pozvánka bude obsahovat odkaz, který bude uživatel odkázán na stránku. Odkaz bude obsahovat ID auta, email a klíč pro spojení, který se vygeneroval stejným způsobem jako klíč pro ověření uživatele v podkapitole 5.2.2. Spolu s odesláním se také uloží záznam do tabulky *car\_connect*.

Po kliknutí na odkaz v pozvánce se stránka přesměruje do aplikace, která rozezná, že se jedná o propojení uživatele s autem a přesměruje pohled na *car\_connect.php*. Tady se napřed ověří platnost zadaného klíče a jestliže auto již nebylo uživateli přiřazeno. Následně aplikace podle e-mailu zjistí jeho existenci v systému. Pokud nebyl doposud registrovaný, ponechá záznamy v tabulce nezměněné a odkáže uživatele na registraci, kde bude mít již předvyplněný e-mail, díky uchované hodnoty e-mailu v adrese.

V případě, že uživatel existuje, spustí se funkce *connect\_car\_to\_user*, která v tabulce *car\_connect* poznamená, že uživatel se s autem propojil a zároveň přiřadí do tabulky *cars\_aggregated* záznam o uživateli s autem.

### 5.2.6 Odeslání upozornění a statistik

O odesílání e-mailu se stará funkce implementována v *daily\_send.php*. Ta se bude spouštět každý den. Pokud server bude mít operační systém Windows bude se spouštět pomocí *plánovač úloh*. V tomhle případě aplikace poběží na linuxovém prostředí na webovém hostingu Endora<sup>7</sup> a bude spouštěna *cron*<sup>8</sup> úloha. Po spuštění se napřed postupně všem uživatelům odešle upozornění pomocí funkce *run\_warning\_send* a poté funkce *stats\_run* pro statistiky.

Tyhle funkce budou následně více popsány:

<sup>7</sup><https://endora.cz>

<sup>8</sup>Časový plánovač úloh v Unixových operačních systémech

## Upozornění

Upozornění se může odeslat dvojím způsobem. První způsob je plánované spuštění, které bylo popsáno výše. Dalším způsobem je pomocí tlačítka, které je umístěno na stránce nastavení v sekci upozornění. V obou případech bude funkce potřebovat ověření klíče uživatele, který dostane jako argument. Uvnitř funkce napřed ověří totožnost uživatele pomocí klíče, a následně frekvenci s kterou se má upozornění odeslat. Zvolená frekvence pro uživatele je uložena v tabulce **users** ve sloupci **report\_warning**. Jestliže nebude nastavená, aplikace neprovede žádnou akci. Podle dané frekvence vybere pouze aktivní záznamy z tabulky *car\_warnings* od všech vozidel, která patří k ověřenému uživateli o odešle e-mail.

## Statistiky

Princip spouštění funkce pro odeslání statistik je téměř totožný s upozorněním, a proto budou vylíčeny jenom implementační detaily této funkce.

Funkce porovná kdy byl poslední e-mail odeslaný a kdy se má odeslat další. Totožně jak je tomu u upozornění, zvolená frekvence je uložena v tabulce **users** navíc obsahuje také sloupec s informací, kdy byl záznam naposledy odeslán. Pokud záznam měl být odeslán a nebyl, spustí se funkce *stats\_run*, která vygeneruje e-mail, který bude obsahovat data z agregačních tabulek pro spotřebu a opravy. Vybere pouze ty záznamy, která patří do daného intervalu podle určené frekvence zasílání. Až po úspěšném odeslání si aplikace poznačí, že se záznam odeslal.

## Kapitola 6

# Testování

Následná kapitola bude věnována testování aplikace. Jelikož pro vývoj aplikace nebyl použit žádný z existujících frameworků, musel být zvolen důkladnější systém pro testování. Testování nakonec probíhalo ve dvou fázích. První fáze, prováděla testování s pomocí automatizovaných testů napsaných v *Robotframeworku*<sup>1</sup>. Druhou fází tvořili fyzické osoby, které mohly být potencionálními uživateli aplikace. Obě fáze budou následně popsány podrobněji.

### 6.1 Automatizace

Automatizované testování probíhalo pomocí zmíněného RobotFrameworku spolu s nástrojem Selenium<sup>2</sup>. Testování bylo rozděleno do několik částí. Jednotlivé části, které budou popsány se vyvíjely během implementace aplikace. Tahle varianta testování byla zvolena z důvodů udržení aplikace ve validním tvaru již od začátku implementace tak, aby později se předešlo nefunkčnosti některých částí, které již byly implementovány a měly by správně fungovat. Nyní budou popsány základní části a na konci jejich celkové zhodnocení.

#### Registrace, přihlášení a potvrzení

Zautomatizování registrace a přihlašování může být prakticky považováno za základ, jelikož s aplikací se může pracovat až po jejich provedení. Automatizační skript imituje uživatele, který přistoupí na stránku a založí si nový účet. Při registraci ověřuje správnou formu včetně chybových hlášek:

- Vyplnění povinných a nepovinných údajů
- Opakované heslo
- Špatný formát e-mailu
- E-mail, který již v databázi existuje

Následně bude pokračovat přihlášení, ve kterém se skript pokusí o přihlášení pod neexistujícím uživatelem, uživatelem se špatným heslem a nakonec se správnými údaji.

Testování potvrzení bude probíhat po úspěšné registraci, kdy se skript přihlásí do e-mailového portálu a otevře příchozí zprávu, kterou obdrží po registraci a klikne na potvrzovací klíč. Test obsahuje taktéž ověření špatného klíče. Tahle varianta je jednodušší,

<sup>1</sup>Testovací framework napsaný v jazyku Python <http://robotframework.org>

<sup>2</sup>Nástroj pro testování webových aplikací <https://www.seleniumhq.org>



protože stačí přistoupit do aplikace, která obsahuje v adrese špatný klíč. V přiloženém výpisu kódu 6.1 je ukázka automatizačního skriptu pro přihlášení. V tomto výpisu se napřed provede načtení argumentů do proměnných, poté se aktivně čeká na stránku 10 sekund, dokud nebude obsahovat tlačítko *submit*. Následně se vloží do tagů *<input>* s názvem *email* a *password* hodnoty z parametrů. V posledním kroku se klikne na tlačítko potvrzení.

|                                  |                            |                       |
|----------------------------------|----------------------------|-----------------------|
| Login                            |                            |                       |
| [Arguments]                      | <code>\${nick}</code>      | <code>\${pass}</code> |
| Wait Until Page Contains Element | <code>name=submit</code>   | <code>10</code>       |
| Input Text                       | <code>name=email</code>    | <code>\${nick}</code> |
| Input Text                       | <code>name=password</code> | <code>\${pass}</code> |
| Click Element                    | <code>name=submit</code>   |                       |
| sleep                            | <code>3 s</code>           |                       |

Výpis 6.1: Ukázka výpisu fungování automatizovaného skriptu

## Ověření formulářů

Protože ověřování formulářů je velmi podobné, budou zde popsány jen ty nejdůležitější položky pro otestování.

Skript se snaží chovat jako běžný uživatel, který nezadal požadované údaje nebo přidal špatný formát obrázku. Tímhle se ověří správné zobrazení hlášek, a výsledně zobrazená data. Za správné chování aplikace se může považovat validní zobrazení výsledných dat, jako například:

- Po editaci se data skutečně změnil.
- Přidání nového upozornění, přidá nový záznam, který bude mít správnou dobu expirace.
- Přidaný záznam ke spotřebě vozidla skutečně přepočítá průběrnou spotřebu.
- Po přidání opravy se přepočítá celková částka oprav a změnil se hodnota v grafu daného měsíce.

## Zhodnocení automatizace

Zvolení automatizovaného testování již od začátku implementace přineslo několik užitečných poznatků, na které se při návrhu a následné implementaci nemyslelo. Jedním z poznatků je automatický posun na vyskakovací okno v případě, že není viditelné v aktuálním pohledu stránky. Celkový postup při implementaci byl časově náročnější, ale zato minimálně v pěti případech dokázaly testy odhalit nefunkčnost již implementované funkce.

## 6.2 Uživatelské testování

Druhá fáze byla uskutečněná až na funkční aplikaci, kde byly zvoleny techniky přímého a nepřímého pozorování. Jednotlivé postupy budou popsány a nakonec zhodnoceny výsledky pozorování.

## Přímé pozorování

Přímé pozorování mělo následující průběh. Tester dostal úkol, který měl vyplnit. Během provádění úkonu jsem si zapisoval poznámky ohledně chování testera, rychlosti provádění úkolů a délku zorientování se na stránce.

## Nepřímé pozorování

Testování probíhalo na skupince lidí, ke kterým nebyl přístup, proto nemohlo dojít k ovlivnění jejich činnosti. Pomocí **google formuláře**<sup>3</sup> bylo testerům poskytnuto několik jednoduchých úkolů, které měli splnit. Po splnění jednotlivých úkolů měli vyplnit několik jednoduchých dotazů na obtížnost, časový horizont úkolu a zda by něco změnili. Poslední otázky ve formuláři byly zaměřeny na intuitivnost aplikace a na celkový pocit testerů při plnění úkolů.

## Zhodnocení pozorování

Během zpracování výsledků se objevilo několik užitečných výstupů na jejichž základě byla aplikace vizuálně i funkcionálně upravena. Konkrétně se jednalo o reakce na:

- Sjednocení všech funkčních tlačítek to jednoho stylu.
- Přidání tlačítek pro přidání nové hodnoty u grafů v pohledu auta.
- Barevně označit již aktivní upozornění.
- Barevně odlišit navigační menu v levé části aplikace. Světlé menu působí ztraceně vzhledem k stránce.

---

<sup>3</sup><https://goo.gl/forms/YW00sgYSAsm6SvII2>

## Kapitola 7

### Závěr

Na začátku práce bylo nutné nastudovat techniky tvorby webových aplikací (PHP, CSS, HTML, JS). Následně cílem bylo navrhnout a implementovat informační systém, který bude sloužit jako technický deník pro motorová vozidla. Jednotlivé specifikace požadavků určené v kapitole o analýze požadavků 3 byly implementovány. Pro rychlejší načítání dat byly vytvořeny agregační tabulky, které obsahují předzpracované hodnoty pro specifické pohledy. Celá aplikace byla testována ve dvou částech. První část se konala v průběhu implementace pomocí automatizovaných skriptů, které bylo nutné nastudovat. Poměrně dost času bylo věnováno na jejich implementaci a vyhodnocování, ale ve výsledku testy ušetřily poměrné množství času zpětnému opravování kódu. Druhá část probíhala pomocí skupiny testerů jejichž poznatky byli zapracovány do aplikace. Podrobnější informace jsou popsány v kapitole 6.

Ve srovnání s produkty, které jsou na trhu, vytvořená aplikace poskytuje příjemné uživatelské rozhraní včetně sdílené správy vozidel. Avšak pro reálné používání u uživatelů, kteří nebudou mít čas se přihlašovat na počítači do aplikace je nutné implementovat k webovému rozhraní i mobilní aplikaci, která bude podporovat také off-line správu. Vytvořená aplikace byla navržena s responzivním rozložením stránky, aby aspoň z části umožnila uživateli přidání údajů ze svého mobilního zařízení. Avšak mobilní aplikaci to nenahradí, což může být dobrým krokem pro rozšíření aplikace.

Návrhu na rozšíření může být nespočet. Jedním z nich je rozšíření o možnost přidání veřejného profilu auta. Auto bude viditelné v samostatném pohledu pro ostatní uživatele, kteří budou moci vozidlu dávat hodnocení. K této možnosti se váže i další možnost rozšíření, kdy aplikace bude brát v potaz periodické přidávání informací o vozidle a takový uživatel bude dostávat ohodnocení od systému. Tohoto ohodnocení bude moci uživatel využít při případném prodeji auta, kdy může k inzerátu přiložit odkaz na jeho profil v *on-line technickém deníku* a což případnému kupujícímu zprostředkuje důvěryhodnější pohled na vozidlo.

# Literatura

- [1] Cyroň, M.: *CSS kaskádové styly: praktický manuál*. Grada, 2006, ISBN 978-802-4714-202.
- [2] Gilfillan, I.: *Myslíme v MySQL 4: knihovna programátora*. Grada, 2003, ISBN 978-802-4706-610.
- [3] Hruška, T.; Křivka, Z.: *Informační systémy (IIS,PIS), Pojem informačního systému, Data, Procesy, Transakce*. [Online].  
URL <https://www.fit.vutbr.cz/study/courses/WAP/private/podklady/Opory/>
- [4] Leiss, O.; Schmidt, J.: *PHP v praxi: pro začátečníky a mírně pokročilé*. Grada, 2010, ISBN 978-802-4730-608.
- [5] Žára, O.: *JavaScript: programátorské techniky a webové technologie*. Computer Press, 2015, ISBN 978-80-251-4573-9.
- [6] Slavoj, P.: *HTML: začínáme programovat*. Grada, 2010, ISBN 978-802-4731-179.

## Přílohy

# Příloha A

## Obsah CD

Přiložené CD obsahuje:

- **\doc**
  - Text bakalářské práce ve formátu PDF
  - Zdrojové soubory textu bakalářské práce
- **\app**
  - Zdrojové soubory aplikace

## Příloha B

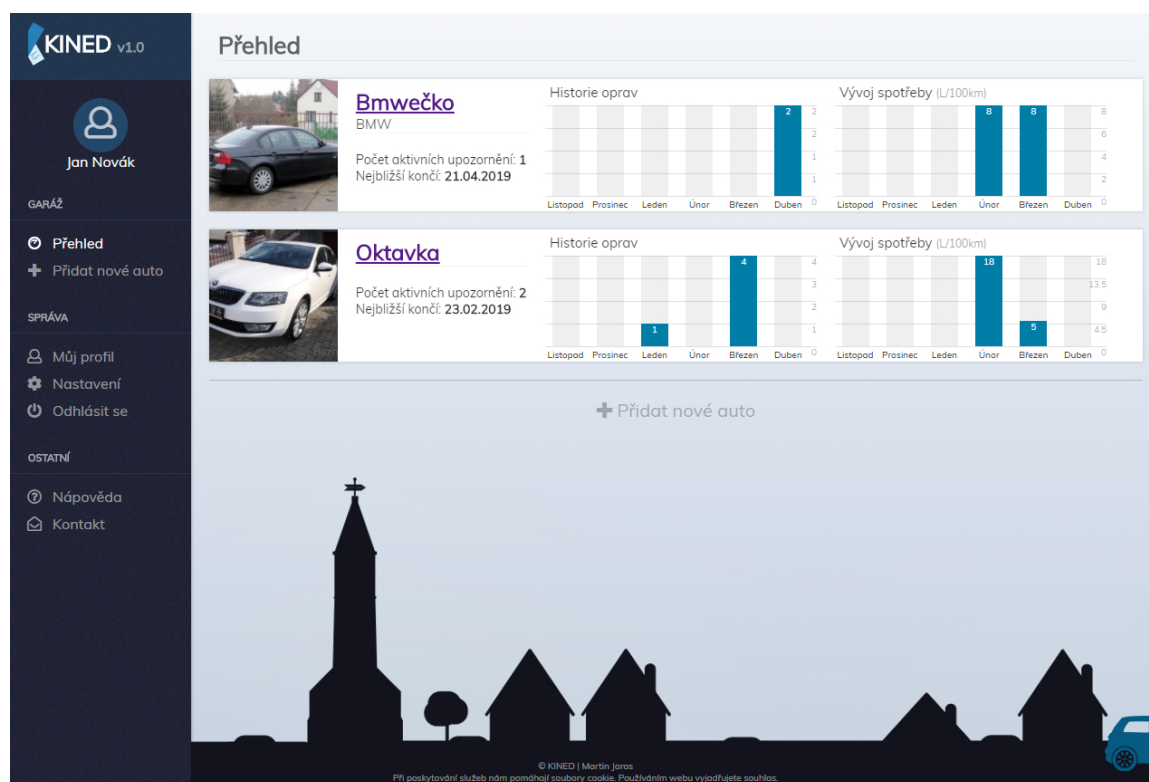
# Testovací protokol

Jako testovací protokol byl použit google formulář <https://goo.gl/forms/YW00sgYSAsm6SvII2>, který obsahoval 9 jednoduchých funkcionálních otázek a 3 obecné otázky. Následuje přehled otázek:

1. Provedte registraci
2. Přihlášení k nově vytvořenému účtu
3. Přidání auta do garáže
4. Přidat upozornění
5. Přidání údajů o spotřebě
6. Přidání opravy
7. Změnit přihlašovací heslo
8. Změnit frekvenci odesílání statistik a odeslat aktuální statistiky
9. Přiřadit firemní auto
10. Přidali nebo odebrali by jste něco?
11. Jaký jste měli pocit při plnění úkolů z aplikace ?
12. Celkové hodnocení provedení

## Příloha C

# Náhled aplikace



Obrázek C.1: Přehled vozidel



Spotřeba
+ Přidat nový záznam

Přidat nový záznam

Zadejte cenu, množství a stav tachometru po natankování. Pokud jste nezadal/a stav tachometru při přidání auta do garáže, udělejte tak prosím nyní.

|  |                  |                   |
|--|------------------|-------------------|
| Cena (Kč)                              | Cena za L        | 30.8 Kč           |
| 2005                                   | Ujetá vzdálenost | 860 km            |
| Množství (L)                           | Cena za Km       | 2.3 Kč/km         |
| 65                                     | Spotřeba         | 7.6 L/100km       |
| Natankování dne                        |                  |                   |
| 04 / 30 / 2018                         |                  |                   |
| Stav tachometru (km)                   | nebo             | Počet najetých km |
| 133160                                 | 860              |                   |
| Stav tachometru před natankováním (km) |                  |                   |
| 132300                                 |                  |                   |

Přidat

Obrázek C.2: Ukázka vyskakovacího okna

Upozornění 2018
Přejít na web

KINED

Online technický deník pro motorová vozidla

Blížící se doba expirace Vašich upozornění

Oktavka (Škoda)

Olej, vyprší za 4 dny (04.05.18)

Dálniční známka, vyprší za 17 dní (17.05.18)

© Copyright 2018, KINED

Obrázek C.3: Ukázka e-mailu s upozorněním